遅延微分方程式の級数による解法 (Computer Algebra : Algorithms, Implementations and Applications)

AUTHOR(S):

平山, 弘; 佐藤, 創太郎

数理解析研究所講究録 1295 巻 2002 年 51-55

遅延微分方程式の級数による解法

平山 弘 (Hiroshi Hirayama)* 神奈川工科大学
佐藤 創太郎 (Soutarou Satou)† 神奈川工科大学

1 はじめに

関数を Taylor 級数展開することは、すでに多くの人によって研究されている自動微分と呼ばれる方法 [12] と数学的には同じものである。この自動微分を利用した常微分方程式の数値解法については、伊理、小野、戸田 [7][10] によって、詳しく論じられている。これら論文では、自動微分を利用して、微分係数が得られるので、これを利用すると Runge-Kutta 法を改良することができることが示されている。しかしながら、この方法は、まだ一般化していないプリ・プロセッサの存在を仮定している。また、高次の公式を得るには、通常の Runge-Kutta の公式 [14][15] と同様にかなり難しいにように思われる。

常微分方程式の解を、任意の次数まで Taylor 級数展開できることは、数学的には、常微分方程式の級数展開による解法と同じであり、その可能性については、久保田 [8] などで述べられており、Corliss and Chang[1] によって、任意の次数の Taylor 級数展開式を利用した常微分方程式を解くためのパッケージ・ソフトウエアが発表されている。Corliss and Chang のソフトウエアは、上記の自動微分と同じようにプリ・プロセッサを利用しているパッケージ・ソフトウエアである。

著者等 [15] は、プログラムでよく使われる演算子（+,-,*,/など）を、被演算の型が異なる場合、別の意味を与えることができる機能、Fortran 90 や C++言語 [2] の機能（オペレータ・オーバロード、operator overload）を使い、有限項で打ち切った Taylor 級数間の四則演算、Taylor 級数の関数演算を定義する。この機能を使うと、プログラムの形で与えられた任意の関数を Taylor 級数展開することができる。常微分方程式の初期値問題の解を任意次数の Taylor 級数展開の形で、容易に得られる。得られた Taylor 級数解を数値計算に利用すれば、任意の次数の数値解法が得られる。この Taylor 級数展開をある次数の Padé 展開式に変形すると、任意次数の A 安定な数値計算方法を与えることを示した。

この計算方法は、微分方程式の数値解法 [9] に、よく使われる陽的 Runge-Kutta 法（RK 法）と比較すると、計算次数が任意とれる点や A 安定な計算法であるため、計算のステップサイズを大きくとっても不安定性が生じない等の利点がある。

陰的 Runge-Kutta 法（IRK 法）と比較すると、A 安定な計算法となるが、IRK 法は計算式が一般に非線形連立方程式となるため、各計算ステップ毎に非線形連立方程式を解く必要がある。Newton 法等の反復法が必要になるので計算が複雑になり、計算時間も多くかかる。

A 安定な計算法は、陽的計算法ではあり得ないとの証明があるが、この計算法は、Padé 展開するために連立一次方程式を解くことが陽的計算となっている。この連立一次方程式は、一般的な連立一次方程式ではなく、特別な形をした Toeplitz 型となる。この連立一次方程式は、通常の連立一次方程式は、未知数の数を n とすると $O(n^3)$ の計算量が必要であるが、Toeplitz 型は Levinson-Durbin の方法 [11] を使えば、$O(n^2)$

*hirayama@sd.kanagawa-it.ac.jp
†sou@pro.sd.kanagawa-it.ac.jp

の計算量で計算できる。さらに、$O(n\log n)$ の計算量で計算できることが知られている。この事実を考慮すれば、ほぼ陽的な計算法と変わらない計算で任意次数の A 安定な計算ができることがわかる。

本論文では、この計算法を遅延微分方程式 [3] に適用することを提案する。ここで提案する計算法は、現在最もよく使われるプログラム言語の一つであり、入手が容易な C++言語を使う。このような計算方法は、数値計算でよく使われる Fortran 90 でも可能である。オペレータ・オーバロードの機能がない C 言語や Fortran 77 では、このような計算は不可能ではないが、現在までこのような計算が行われていない事実を見ると、実際上不可能と思われる。

2 Taylor 展開

この節では、関数を Taylor 展開するための基本的な考え方を説明し、その計算方法について簡単に論じる。詳しくは、Rall[12]、 Henrici[4] や平山 [5] などに述べられている。

2.1 Taylor 級数の四則演算

Taylor 級数の四則計算のプログラムは、以下のように簡単に作ることができる。平行移動によって、展開位置を原点へ移すことができるので一般性を失うことなしに、原点で展開した式だけを扱うことができる。この級数を次のように定義する。

$$f(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + f_4 x^4 + \cdots \tag{1}$$

$$g(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + g_4 x^4 + \cdots \tag{2}$$

$$h(x) = h_0 + h_1 x + h_2 x^2 + h_3 x^3 + h_4 x^4 + \cdots \tag{3}$$

2.1.1 加減算

$h(x)$ が $f(x)$ と $g(x)$ の和差のとき、f, g および h の係数は、次のような関係になる。

$$h(x) = f(x) \pm g(x) \qquad h_i = f_i \pm g_i \tag{4}$$

2.1.2 乗算

$h(x)$ が $f(x)$ と $g(x)$ の積のとき、f, g および h の係数は、次のような関係になる。

$$h(x) = f(x)g(x) \qquad h_n = \sum_{k=0}^{n} f_k g_{n-k} \tag{5}$$

2.1.3 除算

$h(x)$ が $f(x)$ と $g(x)$ の商のとき、すなわち

$$h(x) = \frac{f(x)}{g(x)} \tag{6}$$

であるとき、f, g および h の係数には、以下のような関係が成り立つ。

$$h_0 = \frac{f_0}{g_0} \qquad h_n = \frac{1}{g_0}\left(f_n - \sum_{k=0}^{n-1} h_k g_{n-k}\right) \tag{7}$$

(7) の式は、(6) において、両辺に を掛け、 (1)、(2)、および (3) を代入して、展開する。両辺の同じ次数の係数が等しいことを利用して得られる。

2.2 Taylor 級数の関数

他の多くの初等関数でも同様に係数間の漸化式を導くことができる。ここでは、Taylor 級数の指数関数の計算方法を示す。この方法は後に述べる常微分方程式の解の Taylor 展開の単純な例にもなっている。$h(x) = e^{f(x)}$ とおくと、

$$\frac{dh}{dx} = h\frac{df}{dx} \tag{8}$$

である。(8) に (1)、(3) を代入して、両辺の係数を比較することによって、次のような関係が得られる。

$$h_0 = e^{f_0} \quad h_n = \frac{1}{n}\sum_{k=1}^{n} k h_{n-k} f_k \quad (n \geq 1) \tag{9}$$

対数関数や三角関数の場合も同様な公式を得ることができる。式 (9) のように、Taylor 展開式の指数関数の計算には、指数関数の計算は、1 回しか入らないので、Taylor 展開の指数関数の計算量は、通常の四則演算とそれほど大きな違いにはならない。このことは、対数関数や三角関数などでも同様である。

3 遅延微分方程式の解の Taylor 展開

この節では、Taylor 展開の方法を利用して、遅延微分方程式の解を任意の次数まで Taylor 展開する方法を示す。Taylor 展開された解は、任意次数の一段公式として、Runge-Kutta 公式の代わりとして利用できる。

次のように $\frac{d\mathbf{y}}{dx}$ について解かれた形になっている遅延微分方程式について考える。

$$\frac{d\mathbf{y}}{dx} = \mathbf{f}(x, \mathbf{y}(x), \mathbf{y}(x-\tau)) \tag{10}$$

ここで、$\mathbf{y}$ および $\mathbf{f}$ は、一般にベクトルである。初期条件は

$$\mathbf{y}(x) = \mathbf{g}(x) \qquad x_0 \geq x \geq x_0 - \tau \tag{11}$$

で与えられる。この微分方程式の解の Taylor 展開は、以下に示す Picard の逐次計算法 [13] を使うことによって計算する。

$$\mathbf{y}_n = \mathbf{y}_0 + \int_0^x \mathbf{f}(x, \mathbf{y}_{n-1}(t), \mathbf{y}(t-\tau))dt \tag{12}$$

ただし、(12) の計算は n 次以上の高次の項は省略して計算する。この計算において、$\mathbf{y}(t-\tau)$ は、過去の解であるから、既知関数である。補間計算等を行わないようにするため、計算ステップサイズとして、τ を整数個に分割した大きさをとる。このようにとれば、補間計算を行う必要がないため、誤差が入りにくくなるため、計算精度の向上につながる。この積分を何回も繰り返すことによって、解の Taylor 展開式を得ることが出来る。得られた Taylor 展開を利用して、次の計算位置の定数項を計算する。この定数項を初期値

として、(12) の計算を繰り返すことによって、次の計算位置での Taylor 展開を得る。この操作を繰り返すことによって、微分方程式を解くことができる。

Taylor 展開を利用して、次の位置における定数項の計算を行う場合、この Taylor 展開式を Padé 展開式に変換し、それを利用して、次の位置における定数項を計算すると A 安定な計算法になる。以下の手順をまとめると次のようになる。

(1) 計算ステップサイズを τ の整数分の 1 の大きさにする。

(2) 初期値関数 $\mathbf{g}(x)$ を利用して、微分方程式の初期値 ($\mathbf{y}_0$) を計算する。

(3) (12) を使って、Taylor 展開式を求める。

(4) 得られた Taylor 展開式を可能ならば、Padé 展開式に変換する。

(5) この Padé 展開式または Taylor 展開式を利用して、次の計算 Taylor 展開位置の定数項を計算する。

(6) さらに計算する場合、(3) に戻る。

4 数値例

厳密解が知られている、次の方程式 [16] を解く。$\varphi(x) = exp(2 + cos(x)^2)$ のとき

$$\begin{aligned} \frac{dy}{dx} &= -y(x-1)\left(1+y(x)^2\right) + \varphi(x-1)\left(1+\varphi(x)^2\right) + \varphi'(x), \qquad 2 \geq x \geq 0 \\ y(x) &= \varphi(x), \qquad 0 \geq x \geq -1 \end{aligned} \tag{13}$$

を解くことを考える。この時の厳密解は、$y = \varphi(x)$ である。この微分方程式を $x = 0$ から、計算ステップを $h = 1/1024$ としてとして解く。$x = 0$ における 6 次の Taylor 展開式を求めると、

$$y = 20.0855 - 20.0855 * x^2 + 16.7379 * x^4 - 5.49829e + 13 * x^6$$

この式は、分子分母 3 次の Padé 展開できないので、この Taylor 展開式を利用して、$x = h$ における $y = 20.0854$ を計算する。この値を利用して、$x = h$ における 6 次の Taylor 展開式を求めると、

$$y = 20.0854 - 0.0655854 * x - 22.6421 * x^2 + 337.936 * x^3 - 33376.9 * x^4 + 2.63471e + 06 * x^5 + 6.01748e + 07 * x^6$$

これを $y = num/den$ と Padé 展開すると

$$num = 20.0855 - 2814.47 * x - 105800 * x^2 + 6.84746e + 06 * x^3 den = 1 - 140.124 * x - 5266.46 * x^2 + 340720 * x^3$$

となる。この Padé 展開式を利用して、$x = 2h$ における y の値を計算すると $y = 20.0854$ となる。$x = 2h$ における 6 次の Taylor 展開式を求めると、

$$20.0854 - 0.0655854 * x - 22.6421 * x^2 + 337.936 * x^3 - 33376.9 * x^4 + 2.63471e + 06 * x^5 + 6.01748e + 07 * x^6$$

となる。6 次の係数を見ると非常に大きな係数になっているが、計算ステップ十分に小さいので Padé 展開を利用しないでも十分高精度で計算できる。この計算を 1 万ステップ計算を進めても、6 桁以上の精度を保ち非常に安定した計算ができる。

5 まとめ

本論文では、Taylor 展開を利用して、遅延方程式を解く方法を提案した。この方法は通常の微分方程式を解く場合と同じように、高精度で安定した計算が可能でる。

遅延微分方程式では、硬い（Stiff）状態が発生し、その場合本方法が有用であることを示すことができると期待したが、残念ながら、そのような例を挙げることができなかった。

この方法と他の方法を比較し、どのような場合、どちらの計算方法が良いか判断すべきであるが、厳密解がわかっているような問題では、その差が現れるような現象が見受けられなかった。

いろいろな問題を調べ、本方法を優劣を調べることがこれからの課題である。

参 考 文 献

[1] Corliss G. and Chang Y. F., Solving Ordinary Differential Equations Using Taylor Series, ACM Trans. Math. Soft., Vol.8, pp. 114–144 (1982)

[2] Ellis M. A. and Stroustrup B., The Annotated C++ Reference Manual, Addison-Wesley,(1990)

[3] Hairer E., Wanner G., Solving Ordinary Differential Equations I, Springer-Verlag, (1987)

[4] Henrici, P., Applied Computational Complex Analysis, Vol. 1, John Wiely & Sons, New York,Chap. 1, (1974)

[5] 平山弘, C++言語によるべき型特異点をもつ関数の数値積分, 日本応用数理学会論文誌, Vol.5, pp. 257–266 (1995)

[6] 平山, 小宮, 佐藤, Taylor 級数法による常微分方程式の解法, 日本応用数理学会論文誌, Vol.12, (2002) 出版予定

[7] 伊理, 小野, 戸田, 合成関数の高速微分法とその導関数を含む Runge-Kutta 系の常微分方程式 数値解法公式への応用, 情報処理学会論文誌, Vol.26, pp. 389–396 (1986)

[8] 久保田光一, コンピューテングの玉手箱 続・自動微分 Taylor 展開, bit, Vol.22, pp. 860–861 (1991)

[9] 三井斌友、数値解析入門、朝倉書店、(1985)

[10] 小野, 戸田, 伊理, 微分係数を用いた埋め込 み型 Runge-Kutta 系 2 段公式について, 情報処理学会論文誌, Vol.28, pp. 807–814 (1987)

[11] Press W.H., Flannery B.P., Teukolsky S.A., Vetterling W.T., Numerical Recipes in C, Cambridge University Press, Cambridge, (1988) （邦訳：丹慶、奥村、佐藤、小林訳、技術評論社、(1993)）

[12] Rall,L. B. , Automatic Differentiation-Technique and Applications, Lecture Notes in Computer Science, Vol.120, Springer-Verlag, Berlin-Heidelberg-New York(1981)

[13] 佐野理, キーポイント微分方程式, 岩波書店, 東京, (1993)

[14] 田中, 高山, 安定性のよい 9 段数 7 次陽的 Runge-Kutta 法について, 情報処理学会論文誌, Vol.34, pp. 52–61 (1993)

[15] 田中, 高山,2 段数陰的 Runge-Kutta 法について, 情報処理学会論文誌, Vol.36, pp. 226–234 (1995)

[16] 小藤, 梁, 遅延微分方程式の半群的解法、常微分方程式の数値解法とその周辺, 名古屋大学ベンチャービジネスラボラトリ, pp. 145–158 (2000)